IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re PATENT application of:

Appellant(s):	Matthew Crewe
Serial No:	10/630,020
Filing Date:	July 30, 2003
Title:	FLEXIBLE INTEGRATION OF SOFTWARE APPLICATIONS IN A NETWORK ENVIRONMENT
Examiner:	Brian P. Whipple
Art Unit:	2152
Docket No.	DYOUP0253US

REPLY BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This is a reply to the Examiner's Answer mailed November 17, 2009, for which a two-month period of response was provided.[1] Accordingly, this Reply Brief is timely filed. Kindly enter the following remarks. Appellant respectfully requests reconsideration of the application, reversal of the claims rejections, and allowance of the pending claims.

[1] The Examiner's Answer mailed on November 17, 2009 was not signed by the second conferee. A second Examiner's Answer was mailed on January 5, 2010. Other than the signature of the second conferee, the second Examiner's Answer is identical to the original Examiner's Answer.

ARGUMENTS

This Reply Brief is directed to the comments and arguments in the "Response to Argument" section of the Examiner's Answer.

1. Appellant's Interpretation of the Model-View-Controller Software Architecture is Consistent with the Specification

Claim 1 is directed to, *inter alia*, a software product comprising computer executable instructions embodied on a computer readable medium that are configured when executed to function as ***a model component in a model-view-controller software architecture***. (Emphasis added). As explained at pages 5-6 of the Appeal Brief, the model component is a part of the model-view-controller software architecture that provides the underlying functionality for a Picture Archiving and Communications System (PACS) network.

A model-view-controller software architecture includes a model component, a view component, and a controller component. The model component defines the fundamental aspects of the functionality of an application, while the view component manages the graphical and/or textual output, and the controller interprets user inputs in order to direct the model and/or view components to respond appropriately.

The Examiner contends that there is no reason for adopting this interpretation. (See Examiner's Answer at pages 9-10). The Examiner seemingly ignores the claim phrase "model-view-controller software architecture", and instead points to various aspects of Brandt as disclosing the "model component", the "view component" and the "controller component", without giving any patentable weight to the context in which the claims are expressly directed—a model-view-controller software architecture.

Applicant's interpretation is supported throughout the specification. Support for applicant's interpretation may be found, for example, in the following passages of the specification:

Model Component

- This approach means that an application provider is able to supply a generic model component encapsulating the functionality of his application to all PACS network providers, and need only write a specific glue bridge for each network provider to accommodate any non-conformant aspects of their network. (Page 18, lines 22-26)

- The model component of the application, i.e., that part which performs the functionality of the application... (Page 13, lines 1-5).

- In one embodiment, the application is provided as a single software component, for example the visualization component of an otherwise stand-alone 3D visualization application, and integrated into the PACS network in a manner consistent with the model-view-controller software architecture. In this scheme, the application provider supplies the model as a self-contained software component. The PACS network provider is then free to design controller and view components consistent with the "look and feel" of his PACS network. The freedom of the PACS network provider is limited only by the need for the controller and view components to be capable of interacting appropriately with the model component (i.e. with the functionality of the software application). For example, while the PACS network provider is able to determine which aspects of the model component are accessible to a user, he cannot add more aspects by simply providing for them in his controller if the model component cannot support the corresponding functionality. (Page 10, line 26 – Page 11, line 14)

- Unlike previous integration schemes, with the model-view-controller approach of the invention, the fundamental underlying technical aspects of an application's functionality are hidden from a PACS network provider, but the network provider is nonetheless free to design his own user interface through his defined controller and view components. This means the PACS network provider can quickly generate a proprietary GUI for interacting with the third-party-supplied software component which encompasses all, or any subset of, the functionality of a stand-alone version of the third-party application. The approach means the application provider does not have to release a low-level version of his application making it more amenable to reverse engineering yet allows the PACS network provider to fully integrate the third-party application in to his network with a user interface consistent with the "look and feel" of his system. This can be done more quickly and with less technical effort than that required to integrate a granular component version of the application. With a single software component model-view-controller approach, the PACS network provider can decide which functions of the third party application a user can perform on data being visualised, for example rotations or window and level manipulation for volume rendering, and how the user interface appears, but he cannot alter the fundamental aspects of the functionality of the application, such as use of the algorithms which govern the rotation or rendering. (Page 11, line 15 – Page 12, line 3).

- The software wrapper W is responsible for ensuring the functional software appears to the PACS network as a single self-contained model component M, consistent with the model-view-controller paradigm (Page 12, lines 20-23).

View and Controller Components

- Whichever model component is invoked, the user is presented with the same user interface, as defined by the view and controller components. Page 19, lines 10-12.

- As described above, the PACS provider can design the respective view and controls components to provide a common "look and feel" to the four applications contained in the four model components. Page 17, lines 26 to 29.

- The user is presented with a user interface defined by the view and controller components. Page 12, lines 2-3.

Based on the foregoing, it is clear that the terms "model component", "view component" and "controller component" should be interpreted in the context of a "model-view-controller architecture" and Appellant's interpretation of the claim terms is fully supported by the specification.[2]

2. The "Model Component" Defines the Fundamental Aspects of the Functionality of the Application

As explained in the Appeal Brief at pages 5-6, the "model component" defines the fundamental aspects of the functionality of an application..." The Examiner contends that he is "unaware" of any portion of the Appellant's specification or claims that limits the model component to such a definition. (Examiner's Answer at page 11).

As an initial matter, the Board's attention is drawn to at least the passages referenced above in Section 1, which relate to the "Model Component". From these passages, it is clear directly from the specification that the model component defines the fundamental functionality of the application, within a model-view-controller software architecture.

[2] The Board is invited to type "model view controller" into an Internet search engine, such as Google, and follow some of the resulting links. For example, the "Wikipedia" entry at http://en.wikipedia.org/wiki/ Model%E2%80%93view%E2%80%93controller clearly sets out the well understood meanings of the relevant terms.

In contrast, the Examiner's interpretation of "model component" ignores the express claim language and the specification. In particular, the Examiner has interpreted the term "model component" outside of the realm of the claimed "model-view-controller software architecture". The Examiner contends that the preference files of Brandt disclose the "model component". This contention is made even though Brandt is not even directed to a model-view-controller architecture, as set forth in the Appeal Brief at page 6, line 20- page 8, line 9.

The Examiner further contends that selection of "screen colors, mouse configurations, web browser parameters, etc." and the system administrator disclosed in Brandt relates to fundamental functionality of the model component. (Examiner's Answer at page 11-12). As explained in the Appeal Brief at page 7, line 21- page 8, line 9, each of these components would be provided by the "view component", which manages the graphical and/or textual output or the "controller component", which interprets user inputs in order to direct the model and/or view components to respond appropriately.

In a model-view-controller software architecture, as recited in claim 1, the types of user interfaces parameters set forth in Brandt's preference files have nothing to do with the model component. Indeed, one purpose of the model-view-controller software architecture is that the model component has nothing to do with how the user interface appears. The user interface in the model-view-controller software architecture is defined by the view and controller components. The specification makes this point clear at page 19, lines 10-12, for example, which states: "Whichever model component is invoked, the user is presented with the same user interface, as defined by the view and controller components".

Thus, not only does Brandt simply not disclose anything that one of ordinary skill in the art would remotely consider to a model-view-controller software architecture, as the term is readily understood in the art and explained as meaning in the specification, any attempt to force Brandt to be seen as some kind of model-view-controller software architecture can, at most, only result in the preference files of Brandt mapping to some aspect of the view or controller components of the architecture. This is because these are the components associated with the user interface presented to the user (see Specification at page 13, lines 2-3) and Brandt's preference files are directed to configuring the user's workstation (See Brandt, col. 5, lines 47-52).

3. The Combination of Brandt and Cooke has not been found to disclose the Claimed Invention

The Examiner contends that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. On pages 8-10 of the Appeal Brief, Appellant merely identifies certain deficiencies with Brandt and Cooke. On page 10 of the Appeal Brief, Appellant specifically states:

> it should be evident that even if Brandt were modified by Cooke, the proposed modification would not result in the subject matter recited in claim 1. In particular, neither of the references, alone or in combination, discloses a software product containing a medical-imaging visualization application, where the software product comprises computer executable instructions embodied on a computer readable medium that are configured when executed to function as a model component in a model-view-controller software architecture, and has an interface having a set of user interface control parameters and a set of data handling parameters, the sets of parameters being chosen to allow flexible integration of the visualization application into a PACS network, as recited in claim 1.

Thus, contrary to the Examiner's argument set forth in Sections 3 and 5 of the Examiner's Answer (Examiner's Answer at pages 12-14), Appellant has identified deficiencies with each of the references individually, as well as argued nonobviousness by attacking the references in combination.

4. Appellant has not Argued for Patentability based on Un-Claimed Features

The Examiner contends that Appellant is relying on features not recited in the rejected claims for purposes of patentability. (See Examiner's Answer at page 13). Appellant respectfully disagrees. Appellant merely identified one problem to which aspects of the claims are directed—to allow a PACS network provider to provide customized view and controller components that are consistent with the look and feel of the PACS network, as opposed to having a look and feel dictated by the provider of the model component, as explained in pages 4-5 of the Appeal Brief.

5. Appellant uses the Conventional Meaning for the terms "high level software component" and "low level software component"

Claim 14 is directed to a method for offering a software application to a PACS network integrator. The software application may be provided as a high-level software component (e.g., as schematically represented in the top portion of Figure 6) or as a plurality of lower-level software components (e.g., as schematically represented in the bottom portion of Figure 6). Providing the software components in this fashion allows the PACS network integrator to choose either the first version for ease of integration at the cost of reduced flexibility, or the second version for improved flexibility but more difficult integration. Support for this construction is set forth on page 19, line 24 – page 21, line 27 of the specification.

The Examiner contends that the broadest reasonably interpretation was given to the claim and the preference files of Brandt disclose such these claim elements. Applicant respectfully submits that the Examiner's interpretation is unreasonable and ignores express claim elements set forth in clam 14.

Claim 14 recites: "providing a first version of the application contained in a high-level software component" and "providing a second version of the application contained in a plurality of lower-level software components". Thus, claim 14 is directed to providing a first version of the application and a second version of the application. In contrast, Brandt discloses hierarchical preference files. One having ordinary skill in the art will readily appreciate that preference files are not application files. Furthermore, Brandt's preference files have not been found to disclose different versions of the application. As such, the Examiner's interpretation of the hierarchy of preference files set forth in Brandt for configuring a user's workstation is contrary to the express language set forth in claim 14.

CONCLUSION

For the reasons given above, and for the reasons presented in Appellant's Appeal Brief, Appellant respectfully requests that the Honorable Board reverse the Examiner's rejections of claims 1-19.

In the event any fee or additional fee is due in connection with the filing of this paper, the Commissioner is authorized to charge those fees to our Deposit Account No 18-0988 (Docket No. DYOUP0253US).

Respectfully submitted,

RENNER, OTTO, BOISSELLE & SKLAR, L.L.P.

By: /Timothy E. Manning/
Timothy E. Manning, Reg. No. 48,964

1621 Euclid Avenue
Nineteenth Floor
Cleveland, Ohio 44115
216-621-1113